

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-010815

(43)Date of publication of application : 14.01.2000

(51)Int.Cl.

G06F 11/28

G06F 9/46

G06F 13/00

G06F 15/16

(21)Application number : 10-191092

(71)Applicant : NEC CORP

(22)Date of filing : 22.06.1998

(72)Inventor : OTANI HIROYUKI

AOKI HISAYUKI

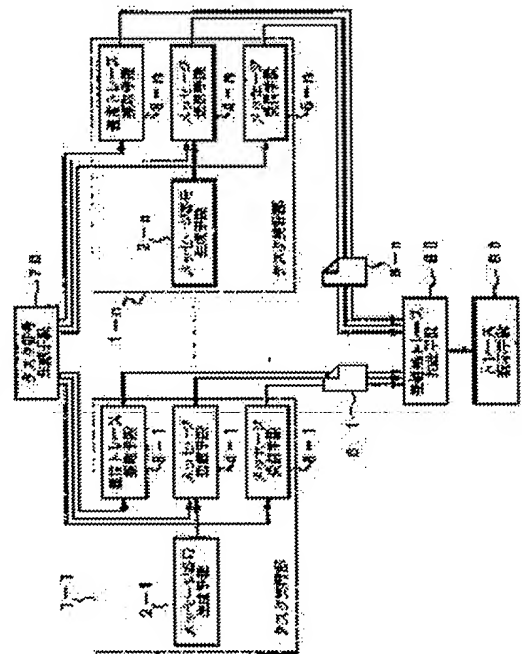
AIBA YUICHI

(54) DEVICE AND METHOD FOR DEBUGGING PARALLEL DECENTRALIZED SYSTEMS

(57)Abstract:

PROBLEM TO BE SOLVED: To actualize an efficient debugging in the debugging for processes operating on parallel decentralized systems.

SOLUTION: In a task 1-1 in a process, a normal trace gathering means 3-1 records identification information of a task number and process contents in a trace file 6-1, a message transmitting means 4-1 embeds a pair of a message number generated by a message number generating means 2-1 and the current task number in a message, sends the pair and also records it in the trace file 6-1, a message receiving means 5-1 records a set of the task number, the message number of a received message and a transmission origin task number in the trace file 6-1 and a trace extracting means 80 specifies the task number and the number of a sent message in the designated task and extracts the trace of the tasks coincident with the received message numbers and the transmission origin task numbers from among the traces.



LEGAL STATUS

[Date of request for examination]

22.06.1998

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

3139459

[Date of registration]

15.12.2000

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-10815

(P2000-10815A)

(43) 公開日 平成12年1月14日 (2000.1.14)

(51) Int.Cl.⁷

識別記号

F I

テマコード* (参考)

G 0 6 F 11/28
9/46
13/00
15/16

3 1 0
3 6 0
3 5 1
4 6 0

G 0 6 F 11/28
9/46
13/00
15/16

3 1 0 A 5 B 0 4 2
3 6 0 B 5 B 0 4 5
3 5 1 N 5 B 0 8 9
4 6 0 D 5 B 0 9 8

審査請求 有 請求項の数13 F D (全 15 頁)

(21) 出願番号 特願平10-191092

(22) 出願日 平成10年6月22日 (1998.6.22)

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 大谷 寛之

東京都港区芝5丁目7番1号 日本電気株式会社内

(72) 発明者 青木 久幸

東京都港区芝5丁目7番1号 日本電気株式会社内

(74) 代理人 100093595

弁理士 松本 正夫

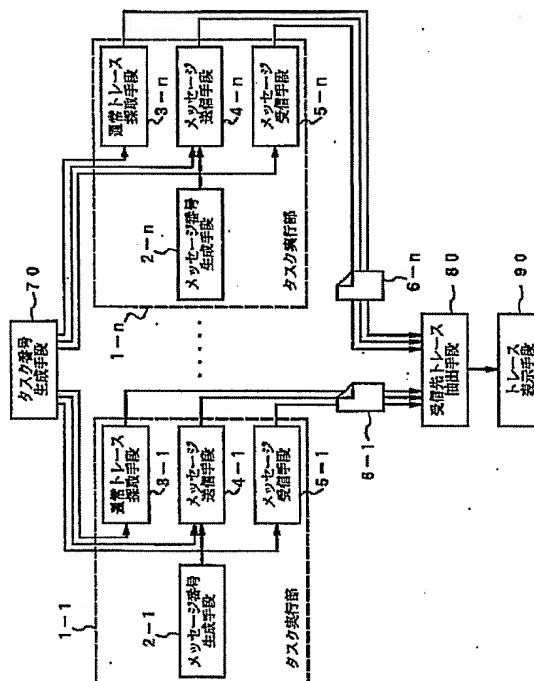
最終頁に続く

(54) 【発明の名称】 並列分散システムのデバッグ装置及びデバッグ方法

(57) 【要約】 (修正有)

【課題】 並列分散システム上で動作するプロセスのデバッグにおいて、効率的なデバッグを実現する。

【解決手段】 プロセス内のタスク1-1において、通常トレース採取手段3-1は、タスク番号と処理内容の識別情報をトレースファイル6-1に記録し、メッセージ送信手段4-1は、メッセージ番号生成手段2-1で生成したメッセージ番号、現在のタスク番号の組をメッセージに埋め込んで送信すると共に、トレースファイル6-1に記録し、メッセージ受信手段5-1は、タスク番号、受信メッセージのメッセージ番号、送信元タスク番号の組をトレースファイル6-1に記録し、トレース抽出手段8-0は、指定されたタスクにおいて、そのタスク番号と送信されたメッセージの番号を特定し、トレースの中から、受信メッセージ番号、送信元タスク番号の一致するタスクのトレースを抽出する。



【特許請求の範囲】

【請求項1】 通信ネットワークを介して複数の計算機が接続され、個々の計算機上でプログラムの実行主体であるプロセスが通信により協調動作する並列分散システムにおいて、

各プロセスが動作する際にプロセス内の一連の処理単位とするタスクにシステム内でユニークなタスク番号を生成するタスク番号生成手段と、

前記プロセス内でユニークなメッセージ番号を生成するメッセージ番号生成手段と、

前記タスク番号生成手段において生成されたタスク番号とプロセスが処理した内容を識別する情報を組にしてトレースに記録する通常トレース採取手段と、

タスクがメッセージを送信する際、前記メッセージ番号生成手段により生成したメッセージ番号と現在のタスク番号とを組にしてメッセージに埋め込むことによりメッセージ送信を行ない、さらに当該メッセージ番号とタスク番号の組をメッセージ送信トレースとして記録するメッセージ送信手段と、

プロセスにメッセージが到着した際に、前記タスク番号生成手段によりタスク番号を生成させ、該タスク番号と受信したメッセージに埋め込まれているメッセージ番号と送信元のタスク番号の組をメッセージ受信トレースとして記録し、到着したメッセージに対応するタスクを実行するメッセージ受信手段とから構成される複数のタスク実行手段と、

指定されたタスクのタスク番号、前記メッセージ送信トレースの前記メッセージ番号、前記メッセージ受信トレースの前記メッセージ番号と送信元のタスク番号に基づいて、所定のタスクを特定することにより、当該タスクのトレースを、記録されたトレースの中から抽出するトレース抽出手段と、

前記トレース抽出手段において抽出したタスクのトレース情報を表示するトレース表示手段とを備えることを特徴とする並列分散システムのデバッグ装置。

【請求項2】 複数の前記タスク実行手段毎に、前記通常トレース採取手段による前記通常トレースと、前記メッセージ送信手段によるメッセージ送信トレースと、前記メッセージ受信手段によるメッセージ受信トレースを記録する複数のトレースファイルを備え、前記トレース抽出手段は、前記トレースファイルからタスクのトレースを抽出することを特徴とする請求項1に記載の並列分散システムのデバッグ装置。

【請求項3】 前記トレース抽出手段が、指定されたタスクのタスク番号と当該タスクから送信したメッセージのメッセージ番号を特定し、さらにメッセージ受信トレースにおいて送信元のタスク番号、受信したメッセージのメッセージ番号が、前記特定したタスク番号、メッセージ番号と一致するタスクのトレースを、記録されたトレースの中から抽出する受信先トレース抽出手段である

ことを特徴とする請求項1又は請求項2に記載の並列分散システムのデバッグ装置。

【請求項4】 前記トレース抽出手段が、特に指定されたタスクがメッセージ受信に対応して実行された場合、メッセージ受信トレースに含まれる送信元のタスク番号を特定し、特定したタスクのトレースを、記録されたトレースの中から抽出する送信元トレース抽出手段であることを特徴とする請求項1又は請求項2に記載の並列分散システムのデバッグ装置。

【請求項5】 前記トレース抽出手段が、特に指定されたタスクのタスク番号とそのタスクから送信したメッセージのメッセージ番号を特定し、さらにメッセージ受信トレースにおいて送信元のタスク番号、受信したメッセージのメッセージ番号が、前記特定したタスク番号、メッセージ番号と一致するタスクのトレースを、記録されたトレースの中から抽出すると共に、抽出したタスクについて前記処理を再帰的に行なうことにより、指定されたタスクからメッセージの送信によって連鎖的に実行された全てのタスクのトレースを抽出する前方トレース抽出手段であることを特徴とする請求項1又は請求項2に記載の並列分散システムのデバッグ装置。

【請求項6】 前記トレース抽出手段が、特に指定されたタスクがメッセージ受信により実行された場合、メッセージ受信トレースに含まれる送信元のタスク番号を特定し、特定したタスクのトレースを記録されたトレースの中から抽出すると共に、特定したタスクについて前記処理を再帰的に行なうことにより、指定されたタスクが実行された起因となるタスクのトレースを全て抽出する後方トレース抽出手段であることを特徴とする請求項1又は請求項2に記載の並列分散システムのデバッグ装置。

【請求項7】 前記トレース抽出手段として、指定されたタスクのタスク番号と当該タスクから送信したメッセージのメッセージ番号を特定し、さらにメッセージ受信トレースにおいて送信元のタスク番号、受信したメッセージのメッセージ番号が、前記特定したタスク番号、メッセージ番号と一致するタスクのトレースを、記録されたトレースの中から抽出する受信先トレース抽出手段と、

特に指定されたタスクがメッセージ受信に対応して実行された場合、メッセージ受信トレースに含まれる送信元のタスク番号を特定し、特定したタスクのトレースを、記録されたトレースの中から抽出する送信元トレース抽出手段と、

特に指定されたタスクのタスク番号とそのタスクから送信したメッセージのメッセージ番号を特定し、さらにメッセージ受信トレースにおいて送信元のタスク番号、受信したメッセージのメッセージ番号が、前記特定したタスク番号、メッセージ番号と一致するタスクのトレースを、記録されたトレースの中から抽出すると共に、抽出

したタスクについて前記処理を再帰的に行なうことにより、指定されたタスクからメッセージの送信によって連鎖的に実行された全てのタスクのトレースを抽出する前方トレース抽出手段と、

特に指定されたタスクがメッセージ受信により実行された場合、メッセージ受信トレースに含まれる送信元のタスク番号を特定し、特定したタスクのトレースを記録されたトレースの中から抽出すると共に、特定したタスクについて前記処理を再帰的に行なうことにより、指定されたタスクが実行された起因となるタスクのトレースを全て抽出する後方トレース抽出手段とを任意に組合わせて構成されることを特徴とする請求項1又は請求項2に記載の並列分散システムのデバッグ装置。

【請求項8】 通信ネットワークを介して複数の計算機が接続され、個々の計算機上でプログラムの実行主体であるプロセスが通信により協調動作する並列分散システムにおいて、

各プロセスが動作する際にプロセス内の一連の処理単位とするタスクにシステム内でユニークなタスク番号を生成するステップと、

前記プロセス内でユニークなメッセージ番号を生成するステップと、

生成されたタスク番号とプロセスが処理した内容を識別する情報を組にして通常トレースとして記録するステップと、

タスクがメッセージを送信する際、生成したメッセージ番号と現在のタスク番号とを組にしてメッセージに埋め込むことによりメッセージ送信を行ない、さらに当該メッセージ番号とタスク番号の組をメッセージ送信トレースとして記録するステップと、

プロセスにメッセージが到着した際に、タスク番号を生成させ、該タスク番号と受信したメッセージに埋め込まれているメッセージ番号と送信元のタスク番号の組をメッセージ受信トレースとして記録し、到着したメッセージに対応するタスクを実行するステップと、

指定されたタスクのタスク番号、前記メッセージ送信トレースの前記メッセージ番号、前記メッセージ受信トレースの前記メッセージ番号と送信元のタスク番号に基づいて、所定のタスクを特定することにより、当該タスクのトレースを、記録されたトレースの中から抽出するステップと、

抽出したタスクのトレース情報を表示するステップとを備えることを特徴とする並列分散システムのデバッグ方法。

【請求項9】 前記トレースの抽出ステップにおいて、指定されたタスクのタスク番号と当該タスクから送信したメッセージのメッセージ番号を特定し、さらにメッセージ受信トレースにおいて送信元のタスク番号、受信したメッセージのメッセージ番号が、前記特定したタスク番号、メッセージ番号と一致するタスクのトレースを、

記録されたトレースの中から抽出することを特徴とする請求項8に記載の並列分散システムのデバッグ方法。

【請求項10】 前記トレースの抽出ステップにおいて、特に指定されたタスクがメッセージ受信に対応して実行された場合、メッセージ受信トレースに含まれる送信元のタスク番号を特定し、特定したタスクのトレースを、記録されたトレースの中から抽出することを特徴とする請求項8に記載の並列分散システムのデバッグ方法。

【請求項11】 前記トレースの抽出ステップにおいて、特に指定されたタスクのタスク番号とそのタスクから送信したメッセージのメッセージ番号を特定し、さらにメッセージ受信トレースにおいて送信元のタスク番号、受信したメッセージのメッセージ番号が、前記特定したタスク番号、メッセージ番号と一致するタスクのトレースを、記録されたトレースの中から抽出すると共に、抽出したタスクについて前記処理を再帰的に行なうことにより、指定されたタスクからメッセージの送信によって連鎖的に実行された全てのタスクのトレースを抽出することを特徴とする請求項8に記載の並列分散システムのデバッグ方法。

【請求項12】 前記トレースの抽出ステップにおいて、特に指定されたタスクがメッセージ受信により実行された場合、前記メッセージ受信トレースに含まれる送信元のタスク番号を特定し、特定したタスクのトレースを記録されたトレースの中から抽出すると共に、特定したタスクについて前記処理を再帰的に行なうことにより、指定されたタスクが実行された起因となるタスクのトレースを全て抽出することを特徴とする請求項8に記載の並列分散システムのデバッグ方法。

【請求項13】 前記トレースの抽出ステップが、指定されたタスクのタスク番号と当該タスクから送信したメッセージのメッセージ番号を特定し、さらにメッセージ受信トレースにおいて送信元のタスク番号、受信したメッセージのメッセージ番号が、前記特定したタスク番号、メッセージ番号と一致するタスクのトレースを、記録されたトレースの中から抽出するステップと、特に指定されたタスクがメッセージ受信に対応して実行された場合、メッセージ受信トレースに含まれる送信元のタスク番号を特定し、特定したタスクのトレースを、記録されたトレースの中から抽出するステップと、特に指定されたタスクのタスク番号とそのタスクから送信したメッセージのメッセージ番号を特定し、さらにメッセージ受信トレースにおいて送信元のタスク番号、受信したメッセージのメッセージ番号が、前記特定したタスク番号、メッセージ番号と一致するタスクのトレースを、記録されたトレースの中から抽出すると共に、抽出したタスクについて前記処理を再帰的に行なうことにより、指定されたタスクからメッセージの送信によって連鎖的に実行された全てのタスクのトレースを抽出するス

テップと、

特に指定されたタスクがメッセージ受信により実行された場合、メッセージ受信トレースに含まれる送信元のタスク番号を特定し、特定したタスクのトレースを記録されたトレースの中から抽出すると共に、特定したタスクについて前記処理を再帰的に行なうことにより、指定されたタスクが実行された起因となるタスクのトレースを全て抽出するステップとを任意に組合わせて構成されることを特徴とする請求項8に記載の並列分散システムのデバッグ方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、通信ネットワークを介して複数の計算機が接続された並列分散システムにおいて、複数のプロセスが通信により協調動作するプログラムのデバッグ装置及びデバッグ方法に関する。

【0002】

【従来の技術】従来より、プログラムのデバッグを行なうための伝統的な手法として、処理の内容を識別する情報を記録する特殊なコードをあらかじめプログラム内に埋め込んでおき、プログラムの実行後、記録された処理の履歴情報から処理の流れを把握するトレース手法がある。

【0003】一方、並列分散システム上で動作するプログラムをデバッグするツールとして、例えば特開平5-224984号公報に開示されるような分散プログラムデバッグ支援装置がある。

【0004】この分散プログラムデバッグ支援装置は、計算機上のプロセス間で送受信処理が行なわれた際、その実行履歴データを採取する手段、実行履歴データを送信する手段、送信された実行履歴データを受信し収集する手段、実行履歴データを因果関係の順に並びかえ、並びかえた実行履歴データの中から送信時の実行履歴データと受信時の実行履歴データを対応させ、プロセスの実行状況を解析する解析手段、解析結果を表示する手段を含む装置から構成される。

【0005】この装置では、プロセス間の送受信処理をイベントとして、イベントが発生した際、そのイベントの内容、イベントの発生時間、相手先のプロセスIDなどを実行履歴データに残し、そのデータを収集して解析を行ない実行されたプロセスの実行状況を視覚的に表示する。これにより、並列分散システム上で動作するプログラムのデバッグを行なう際、プログラムにトレースを残すための特殊なコードを埋め込むことなく、複数のプロセスの実行状況、プロセス間の通信を視覚的に把握することを可能としている。

【0006】

【発明が解決しようとする課題】上述した従来の並列分散システム上のプログラムのデバッグにおいては、以下に述べるような問題点があった。

【0007】第1に、単にプログラム中に処理の内容を識別する情報を記録する特殊なコードを埋め込む伝統的なトレース手法では、各プロセスで処理された個々の内容の妥当性を把握することができるが、プロセス間の処理の流れの把握は困難である。特に、多数のプロセスが協調動作する大規模な並列分散システムでは、その記録されるトレースの量も膨大で、その中から目的とするトレースを探す必要があり、デバッグ効率が悪いという問題があった。

【0008】第2に、前記分散プログラムデバッグ支援装置は、プログラムの改造を必要とせず、通信によるプロセス間の実行履歴の因果関係を把握することができるものの、実際のデバッグでは、プロセスが実行した処理の詳細な内容を把握する必要もあり、プロセス間の実行履歴の因果関係のみでは情報不足で、やはりプログラム内に何らかの情報を埋め込んだトレースが必要になる。

【0009】また、上記分散プログラムデバッグ支援装置では、複数のプロセス間の実行履歴の因果関係を視覚的に表示することが可能であるが、プロセス数が膨大になると、その因果関係の表示にも限界があり、視覚的に把握することも極めて難しくなる。

【0010】そこで、大規模な並列分散システムにおいて効率的なデバッグを行なうためには、多数のプロセスから採取される膨大なトレースデータの中から、デバッグの対象となるトレースを絞ることが必要になる。例えば、プロセス内のある一連の処理単位とするタスクに注目して、そのタスクからの通信によって他のタスクに影響を与えたか、その結果どのような処理が行なわれたか、あるいはそのタスクが実行された原因は何か等を調査する際、対象となるタスクのトレースのみを膨大なトレースデータの中から抽出できればよい。

【0011】本発明の第1の目的は、指定されたタスクからメッセージ送信が行なわれたことにより、それを受けて実行されたタスクのトレース情報のみを記録されたトレースデータの中から抽出することを可能としてデバッグ効率の向上を実現する並列分散システムのデバッグ装置及びデバッグ方法を提供することにある。

【0012】また、本発明の第2の目的は、指定されたタスクをメッセージ送信によって実行させた送信元のタスクのトレースをトレースデータの中から抽出することを可能としてデバッグ効率の向上を実現する並列分散システムのデバッグ装置及びデバッグを提供することにある。

【0013】また、本発明の第3の目的は、指定されたタスクからメッセージ送信が行なわれたことにより、そのタスクから連鎖的に実行された全てのタスクのトレースをトレースデータの中から抽出することを可能としてデバッグ効率の向上を実現する並列分散システムのデバッグ装置及びデバッグ方法を提供することにある。

【0014】さらに、本発明の第4の目的は、指定され

たタスクをメッセージ送信によって実行させた送信元のタスクのトレースを遡って、そのタスクを実行させた根源となるタスクのトレースを抽出することを可能としてデバッグ効率の向上を実現する並列分散システムのデバッグ装置及びデバッグ方法を提供することにある。

【0015】さらに、本発明の第5の目的は、指定されたタスクからメッセージ送信が行なわれたことにより、それを受けて実行されたタスクのトレース、指定されたタスクをメッセージ送信によって実行させた送信元のタスクのトレース、指定されたタスクからメッセージ送信が行なわれたことにより、そのタスクから連鎖的に実行された全てのタスクのトレース及び指定されたタスクをメッセージ送信によって実行させた送信元のタスクのトレースを遡って、そのタスクを実行させた根源となるタスクのトレースを選択的に抽出することを可能とし、これによりデバッグ効率の向上を実現する並列分散システムのデバッグ装置及びデバッグ方法を提供することにある。

【0016】

【課題を解決するための手段】上記目的を達成する本発明の並列分散システムのデバッグ装置は、通信ネットワークを介して複数の計算機が接続され、個々の計算機上でプログラムの実行主体であるプロセスが通信により協調動作する並列分散システムにおいて、各プロセスが動作する際にプロセス内の一連の処理単位とするタスクにシステム内でユニークなタスク番号を生成するタスク番号生成手段と、前記プロセス内でユニークなメッセージ番号を生成するメッセージ番号生成手段と、前記タスク番号生成手段において生成されたタスク番号とプロセスが処理した内容を識別する情報を組にしてトレースに記録する通常トレース採取手段と、タスクがメッセージを送信する際、前記メッセージ番号生成手段により生成したメッセージ番号と現在のタスク番号とを組にしてメッセージに埋め込むことによりメッセージ送信を行ない、さらに当該メッセージ番号とタスク番号の組をメッセージ送信トレースとして記録するメッセージ送信手段と、プロセスにメッセージが到着した際に、前記タスク番号生成手段によりタスク番号を生成させ、該タスク番号と受信したメッセージに埋め込まれているメッセージ番号と送信元のタスク番号の組をメッセージ受信トレースとして記録し、到着したメッセージに対応するタスクを実行するメッセージ受信手段とから構成される複数のタスク実行手段と、指定されたタスクのタスク番号、前記メッセージ送信トレースの前記メッセージ番号、前記メッセージ受信トレースの前記メッセージ番号と送信元のタスク番号に基づいて、所定のタスクを特定することにより、当該タスクのトレースを、記録されたトレースの中から抽出するトレース抽出手段と、前記トレース抽出手段において抽出したタスクのトレース情報を表示するトレース表示手段とを備える。

【0017】請求項2の本発明の並列分散システムのデバッグ装置は、複数の前記タスク実行手段毎に、前記通常トレース採取手段による前記通常トレースと、前記メッセージ送信手段によるメッセージ送信トレースと、前記メッセージ受信手段によるメッセージ受信トレースを記録する複数のトレースファイルを備え、前記トレース抽出手段は、前記トレースファイルから前記タスクのトレースを抽出することを特徴とする。

【0018】請求項3の本発明の並列分散システムのデバッグ装置は、前記トレース抽出手段が、指定されたタスクのタスク番号と当該タスクから送信したメッセージのメッセージ番号を特定し、さらにメッセージ受信トレースにおいて送信元のタスク番号、受信したメッセージのメッセージ番号が、前記特定したタスク番号、メッセージ番号と一致するタスクのトレースを、記録されたトレースの中から抽出する受信先トレース抽出手段であることを特徴とする。

【0019】これにより、並列分散システム上で動作するプログラムのデバッグを行なう際、多数のプロセスから採取されたトレースデータの中から、ある指定されたタスクよりメッセージを受けて実行されたタスクのトレースのみを抽出して表示することができるので、膨大なトレースデータを全て調査することなく対象を絞ることができ、デバッグ効率は大幅に向上する。

【0020】請求項4の本発明の並列分散システムのデバッグ装置は、前記トレース抽出手段が、特に指定されたタスクがメッセージ受信に対応して実行された場合、メッセージ受信トレースに含まれる送信元のタスク番号を特定し、特定したタスクのトレースを、記録されたトレースの中から抽出する送信元トレース抽出手段であることを特徴とする。

【0021】これにより、並列分散システム上で動作するプログラムのデバッグを行なう際、多数のプロセスから採取されたトレースデータの中から、ある指定されたタスクを実行させた元のタスクのトレースのみを抽出して表示することができるので、デバッグ効率は大幅に向上する。

【0022】請求項5の本発明の並列分散システムのデバッグ装置は、前記トレース抽出手段が、特に指定されたタスクのタスク番号とそのタスクから送信したメッセージのメッセージ番号を特定し、さらにメッセージ受信トレースにおいて送信元のタスク番号、受信したメッセージのメッセージ番号が、前記特定したタスク番号、メッセージ番号と一致するタスクのトレースを、記録されたトレースの中から抽出すると共に、抽出したタスクについて前記処理を再帰的に行なうことにより、指定されたタスクからメッセージの送信によって連鎖的に実行された全てのタスクのトレースを抽出する前方トレース抽出手段であることを特徴とする。

【0023】これにより、並列分散システム上で動作す

るプログラムのデバッグを行なう際、多数のプロセスから採取されたトレースデータの中から、ある指定されたタスクが実行されたことが原因で、連鎖的に実行された全てのタスクのトレースを抽出して表示することができるので、デバッグ効率は大幅に向上する。

【0024】請求項6の本発明の並列分散システムのデバッグ装置は、前記トレース抽出手段が、特に指定されたタスクがメッセージ受信により実行された場合、メッセージ受信トレースに含まれる送信元のタスク番号を特定し、特定したタスクのトレースを記録されたトレースの中から抽出すると共に、特定したタスクについて前記処理を再帰的に行なうことにより、指定されたタスクが実行された起因となるタスクのトレースを全て抽出する後方トレース抽出手段であることを特徴とする。

【0025】これにより、注目したタスクを実行させた起因となるタスクのトレースを抽出して表示することができるので、デバッグ効率は大幅に向上する。

【0026】請求項7の本発明の並列分散システムのデバッグ装置は、トレース抽出手段として、指定されたタスクのタスク番号と当該タスクから送信したメッセージのメッセージ番号を特定し、さらにメッセージ受信トレースにおいて送信元のタスク番号、受信したメッセージのメッセージ番号が、前記特定したタスク番号、メッセージ番号と一致するタスクのトレースを、記録されたトレースの中から抽出する受信先トレース抽出手段と、特に指定されたタスクがメッセージ受信に対応して実行された場合、メッセージ受信トレースに含まれる送信元のタスク番号を特定し、特定したタスクのトレースを、記録されたトレースの中から抽出する送信元トレース抽出手段と、特に指定されたタスクのタスク番号とそのタスクから送信したメッセージのメッセージ番号を特定し、さらにメッセージ受信トレースにおいて送信元のタスク番号、受信したメッセージのメッセージ番号が、前記特定したタスク番号、メッセージ番号と一致するタスクのトレースを、記録されたトレースの中から抽出すると共に、抽出したタスクについて前記処理を再帰的に行なうことにより、指定されたタスクからメッセージの送信によって連鎖的に実行された全てのタスクのトレースを抽出する前方トレース抽出手段と、特に指定されたタスクがメッセージ受信により実行された場合、メッセージ受信トレースに含まれる送信元のタスク番号を特定し、特定したタスクのトレースを記録されたトレースの中から抽出すると共に、特定したタスクについて前記処理を再帰的に行なうことにより、指定されたタスクが実行された起因となるタスクのトレースを全て抽出する後方トレース抽出手段とを任意に組合わせて構成されることを特徴とする。

【0027】請求項8の本発明の並列分散システムのデバッグ方法は、通信ネットワークを介して複数の計算機が接続され、個々の計算機上でプログラムの実行主体で

あるプロセスが通信により協調動作する並列分散システムにおいて、各プロセスが動作する際にプロセス内の一連の処理単位とするタスクにシステム内でユニークなタスク番号を生成するステップと、前記プロセス内でユニークなメッセージ番号を生成するステップと、生成されたタスク番号とプロセスが処理した内容を識別する情報を組にして通常トレースとして記録するステップと、タスクがメッセージを送信する際、生成したメッセージ番号と現在のタスク番号とを組にしてメッセージに埋め込むことによりメッセージ送信を行ない、さらに当該メッセージ番号とタスク番号の組をメッセージ送信トレースとして記録するステップと、プロセスにメッセージが到着した際に、タスク番号を生成させ、該タスク番号と受信したメッセージに埋め込まれているメッセージ番号と送信元のタスク番号の組をメッセージ受信トレースとして記録し、到着したメッセージに対応するタスクを実行するステップと、指定されたタスクのタスク番号、前記メッセージ送信トレースの前記メッセージ番号、前記メッセージ受信トレースの前記メッセージ番号と送信元のタスク番号に基づいて、所定のタスクを特定することにより、当該タスクのトレースを、記録されたトレースの中から抽出するステップと、抽出したタスクのトレース情報を表示するステップとを備える。

【0028】請求項9の本発明の並列分散システムのデバッグ方法は、前記トレースの抽出ステップにおいて、指定されたタスクのタスク番号と当該タスクから送信したメッセージのメッセージ番号を特定し、さらにメッセージ受信トレースにおいて送信元のタスク番号、受信したメッセージのメッセージ番号が、前記特定したタスク番号、メッセージ番号と一致するタスクのトレースを、記録されたトレースの中から抽出することを特徴とする。

【0029】請求項10の本発明の並列分散システムのデバッグ方法は、前記トレースの抽出ステップにおいて、特に指定されたタスクがメッセージ受信に対応して実行された場合、メッセージ受信トレースに含まれる送信元のタスク番号を特定し、特定したタスクのトレースを、記録されたトレースの中から抽出することを特徴とする。

【0030】請求項11の本発明の並列分散システムのデバッグ方法は、前記トレースの抽出ステップにおいて、特に指定されたタスクのタスク番号とそのタスクから送信したメッセージのメッセージ番号を特定し、さらにメッセージ受信トレースにおいて送信元のタスク番号、受信したメッセージのメッセージ番号が、前記特定したタスク番号、メッセージ番号と一致するタスクのトレースを、記録されたトレースの中から抽出すると共に、抽出したタスクについて前記処理を再帰的に行なうことにより、指定されたタスクからメッセージの送信によって連鎖的に実行された全てのタスクのトレースを抽

出することを特徴とする。

【0031】請求項12の本発明の並列分散システムのデバッグ方法は、前記トレースの抽出ステップにおいて、特に指定されたタスクがメッセージ受信により実行された場合、前記メッセージ受信トレースに含まれる送信元のタスク番号を特定し、特定したタスクのトレースを記録されたトレースの中から抽出すると共に、特定したタスクについて前記処理を再帰的に行なうことにより、指定されたタスクが実行された起因となるタスクのトレースを全て抽出することを特徴とする。

【0032】請求項13の本発明の並列分散システムのデバッグ方法は、前記トレースの抽出ステップが、指定されたタスクのタスク番号と当該タスクから送信したメッセージのメッセージ番号を特定し、さらにメッセージ受信トレースにおいて送信元のタスク番号、受信したメッセージのメッセージ番号が、前記特定したタスク番号、メッセージ番号と一致するタスクのトレースを、記録されたトレースの中から抽出するステップと、特に指定されたタスクがメッセージ受信に対応して実行された場合、メッセージ受信トレースに含まれる送信元のタスク番号を特定し、特定したタスクのトレースを、記録されたトレースの中から抽出するステップと、特に指定されたタスクのタスク番号とそのタスクから送信したメッセージのメッセージ番号を特定し、さらにメッセージ受信トレースにおいて送信元のタスク番号、受信したメッセージのメッセージ番号が、前記特定したタスク番号、メッセージ番号と一致するタスクのトレースを、記録されたトレースの中から抽出すると共に、抽出したタスクについて前記処理を再帰的に行なうことにより、指定されたタスクからメッセージの送信によって連鎖的に実行された全てのタスクのトレースを抽出するステップと、特に指定されたタスクがメッセージ受信により実行された場合、メッセージ受信トレースに含まれる送信元のタスク番号を特定し、特定したタスクのトレースを記録されたトレースの中から抽出すると共に、特定したタスクについて前記処理を再帰的に行なうことにより、指定されたタスクが実行された起因となるタスクのトレースを全て抽出するステップとを任意に組合わせて構成されることを特徴とする。

【0033】

【発明の実施の形態】以下、本発明の実施の形態について図面を参照して詳細に説明する。図1は、本発明の第1の実施の形態に係る並列分散システムにおけるデバッグ装置の構成を示すブロック図である。

【0034】本デバッグ装置は、並列分散システム上で動作する複数のタスク実行部1-1~1-n、トレースファイル6-1~6-n、タスク番号生成手段70、受信先トレース抽出手段80、トレース表示手段90から構成される。また、各タスク実行部1-1~1-nは、各々メッセージ番号生成手段2-1~2-n、通常トレ

ース採取手段3-1~3-n、メッセージ送信手段4-1~4-n、メッセージ受信手段5-1~5-nから構成されている。

【0035】メッセージ番号生成手段2-1~2-nは、プロセス内でユニークなメッセージ番号を生成する。通常トレース採取手段3-1~3-nは、タスク番号生成手段70において生成されたタスク番号及びプロセスが処理した内容を識別する情報を採取してトレースに記録する。メッセージ送信手段4-1~4-nは、タスクがメッセージを送信する際、メッセージ番号生成手段2-1~2-nを呼び出して生成したメッセージ番号と送信を行なうタスクのタスク番号の組をそのメッセージに埋め込んでメッセージ送信を行ない、さらにそのメッセージ番号、タスク番号の組をメッセージ送信トレースとして記録する。メッセージ受信手段5-1~5-nは、他のプロセスからメッセージが到着した際、タスク番号生成手段70を呼び出してタスク番号を生成し、生成したタスク番号、受信したメッセージに埋め込まれているメッセージ番号、送信元のタスク番号を組にしてメッセージ受信トレースに記録し、メッセージに対応するタスクを実行する。

【0036】トレースファイル6-1~6-nは、各プロセスから採取されたトレースデータを記録する。タスク番号生成手段70は、タスクを実行する際、システム内ユニークなタスク番号を生成する。受信先トレース抽出手段80は、各プロセスから採取されたトレースファイル6-1~6-nを収集し、指定されたタスク番号とそのタスクから送信したメッセージのメッセージ番号を特定し、さらにメッセージ受信トレースの中から送信元のタスク番号、受信したメッセージ番号が、前記特定したタスク番号、メッセージ番号と一致するタスクのトレースを抽出する。トレース表示手段90は、受信先トレース抽出手段80で抽出したタスクのトレースを表示する。

【0037】ここで、タスク番号生成手段70は、システム内ユニークなタスク番号を生成する手段であるが、例えば並列分散システムがUNIX計算機で構築される場合、タスク番号は、その各計算機をユニークに識別するIPアドレスあるいはノード名、各計算機上で動作するプロセスのプロセスID、各プロセスで動作するタスクをユニークに識別するようなタスクIDから構成することもできる。

【0038】図2は、採取されるトレースのデータ構造の一例を表した図である。各トレースのエントリは、トレースを採取したタスクのタスク番号201、トレースデータを識別するデータ識別子202及びデータ部203から構成される。データ識別子202は、そのトレースを記録した手段によりユニークに識別できる値が記録され、そのトレースのエントリが通常トレース、メッセージ送信トレース、メッセージ受信トレースのいずれの

トレースであるか識別できる。データ部203は、各トレースの種類によって異なり、通常トレースの場合、処理内容の識別情報が記録される。メッセージ送信トレースの場合、送信されたメッセージのメッセージ番号が記録される。メッセージ受信トレースの場合、送信元のタスク番号、受信したメッセージのメッセージ番号が記録される。

【0039】本発明における第1の実施の形態の詳細な動作を図に基づいて説明する。図3は、通常トレース採取手段3-1~3-nの動作を示すフローチャートである。プログラムのデバッグを行なう際は、処理されたタスクの内容を識別できる情報、例えば処理内容を示す文字列等をそのタスクが実行されたタイミングでトレースに残すようにトレース採取命令をあらかじめプログラム中に埋め込んでおく。通常トレース採取手段3-1~3-nは、そのトレースを採取する命令が実行された場合に呼び出される。

【0040】各タスク実行部1-1~1-nの通常トレース採取手段3-1~3-nは、まず、現在実行されているタスクのタスク番号を取得する(ステップ301)。このタスク番号は、タスクを実行した時点で既にタスク番号生成手段70によって各タスクに割り振られる番号である。そして、タスク番号とプログラムに埋め込まれた処理内容を識別する情報との組を通常トレースデータとして、対応するトレースファイル6-1~6-nに記録する(ステップ302)。

【0041】図4は、メッセージ送信手段4-1~4-nの動作を示すフローチャートである。メッセージ送信手段4-1~4-nは、タスク内でメッセージ送信が行なわれる際に呼び出される。

【0042】メッセージ送信手段4-1~4-nは、まず、メッセージ番号生成手段2-1~2-nに問い合わせることにより、送信メッセージに対するメッセージ番号を生成する(ステップ401)。次に、メッセージ送信を行なうタスクのタスク番号を入手する(ステップ402)。その後、送信メッセージのメッセージ番号と送信処理を行なうタスクのタスク番号との組をメッセージ内に埋め込んで、メッセージを相手先のプロセスに送信する(ステップ403)。さらに、メッセージ送信が行なわれたことを示すために、送信処理を行なったタスク番号と送信したメッセージのメッセージ番号の組をメッセージ送信トレースデータとして、対応するトレースファイル6-1~6-nに記録する(ステップ404)。

【0043】図5は、メッセージ受信手段5-1~5-nの動作を示すフローチャートである。メッセージ受信手段5-1~5-nは、プロセスがメッセージを受信したことにより、それに対応するタスクを新たに実行する際に呼び出される。

【0044】メッセージ受信手段5-1~5-nは、受信したメッセージに埋め込まれているメッセージ番号及

び送信元のタスク番号を取り出す(ステップ501)。その後、受信したメッセージを受けて実行するタスクのタスク番号をタスク番号生成手段70に問い合わせ生成する(ステップ502)。さらに、メッセージ受信処理が行なわれた事を示すために、メッセージを受けて実行するタスク番号と受信したメッセージのメッセージ番号及び送信元のタスク番号との組をメッセージ受信トレースとして、対応するトレースファイル6-1~6-nに記録する(ステップ503)。そして、受信したメッセージに対応するタスクを実行する(ステップ504)。

【0045】図6は、受信先トレース抽出手段80の動作を示すフローチャートである。受信先トレース抽出手段80は、デバッグを行なう対象となるタスク番号を指定されて処理を実行する。

【0046】まず、受信先トレース抽出手段80は、並列分散システム上で動作する各プロセスから採取したトレースファイル6-1~6-nを収集する(ステップ601)。次に、収集したトレースファイル6-1~6-nからメッセージ送信処理及びメッセージ受信処理を行なったトレースを抽出する(ステップ602)。

【0047】メッセージ送信トレースからは、メッセージ送信処理を行なったタスク番号と送信したメッセージのメッセージ番号を抽出する。一方、メッセージ受信トレースからは、受信処理により実行されたタスク番号、受信したメッセージのメッセージ番号、送信元のタスク番号を抽出し、メッセージ送受信リストを作成する(ステップ603)。

【0048】作成したメッセージ送受信リストを確認し、もし指定したタスクからメッセージ送信処理が行なわれていた場合(ステップ604)、そのタスクから送信したメッセージのメッセージ番号を抽出する(ステップ605)。

【0049】次に、メッセージ送受信リスト中のメッセージ受信トレースの一覧から、送信元のタスク番号及び受信したメッセージのメッセージ番号が、上記指定したタスクのタスク番号及び抽出した送信メッセージのメッセージ番号と一致するタスクを検索する(ステップ606)。そして、上記指定したタスクと検索されたタスクのトレースをトレースファイル6-1~6-nから抽出する(ステップ607)。

【0050】この抽出するトレースには、そのタスク内で処理された内容を識別する情報、そのタスクで行なわれたメッセージの送受信処理のトレースが含まれる。その後、トレース表示手段90は受信先トレース抽出手段80で抽出したタスクのトレースを表示する。

【0051】この結果、並列分散システムで動作するプログラムのデバッグを行なう際、膨大なトレースデータの中から、指定したタスク及びそのタスクからメッセージ送信が行なわれ、そのメッセージを受けて実行された

タスクのトレース情報のみを表示することができるものである。

【0052】次に、本発明における発明の第2の実施の形態に係るデバッグ装置について説明する。図7は、本発明における第2の実施の形態に係るデバッグ装置の構成を示すブロック図である。本実施の形態は、第1の実施の形態と同様にタスク実行部1-1~1-n、メッセージ番号生成手段2-1~2-n、通常トレース採取手段3-1~3-n、メッセージ送信手段4-1~4-n、メッセージ受信手段5-1~5-n、トレースファイル6-1~6-n、タスク番号生成手段70、トレース表示手段90を備え、さらに、第1の実施例の受信先トレース抽出手段80の代わりに、指定されたタスクをメッセージ送信により実行させた送信元のタスクのトレースを抽出する送信元トレース抽出手段100を備えて構成される。

【0053】この第2の実施の形態の詳細な動作を図に基づいて説明する。図8は、送信元トレース抽出手段100の動作を示すフローチャートである。その他の手段の動作については、第1の実施の形態の場合と同様である。

【0054】送信元トレース抽出手段100は、デバッグを行なう対象となるタスク番号を指定されて処理を行なう。まず、送信元トレース抽出手段100は、第1の実施の形態における受信先トレース抽出手段80と同様に、並列分散システム上で動作する各プロセスから採取したトレースファイル6-1~6-nを収集し、収集したトレースファイル6-1~6-nからメッセージ送信処理及びメッセージ送信処理を行なったトレースを抽出してメッセージ送受信リストを作成する（ステップ801~803）。

【0055】作成したメッセージ送受信リストを確認し、もし指定されたタスクがメッセージ受信により実行されていた場合（ステップ804）、そのメッセージを送信した送信元のタスク番号をメッセージ受信トレースから特定する（ステップ805）。

【0056】次に、上記指定されたタスクと特定した送信元のタスク番号に該当するタスクのトレースをトレースファイル6-1~6-nの中から抽出する（ステップ806）。その後、トレース表示手段90は、送信元トレース抽出手段100で抽出したタスクのトレースを表示する。

【0057】この結果、並列分散システム上で動作するプログラムのデバッグを行なう際、膨大なトレースデータの中から指定したタスク及びそのタスクを実行させた元のタスクのトレース情報のみを表示することができる。

【0058】次に、本発明における発明の第3の実施の形態に係るデバッグ装置について説明する。図7は、本発明における第3の実施の形態に係るデバッグ装置の構成を示すブロック図である。本実施の形態は、第1の実施の形態と同様にタスク実行部1-1~1-n、メッセージ番号生成手段2-1~2-n、通常トレース採取手段3-1~3-n、メッセージ送信手段4-1~4-n、メッセージ受信手段5-1~5-n、トレースファイル6-1~6-n、タスク番号生成手段70、トレース表示手段90を備え、さらに、第1の実施の形態の受信先トレース抽出手段80の代わりに、指定されたタスクが実行されたことにより、連鎖的に実行された全てのタスクのトレースを抽出する前方トレース抽出手段110を備えて構成される。

【0059】第3の実施の形態による詳細な動作を図に基づいて説明する。図10は、前方トレース抽出手段110の動作を示すフローチャートである。その他の手段の動作については、第1の実施の形態の場合と同様である。

【0060】前方トレース抽出手段110は、デバッグを行なう対象となるタスク番号を指定されて処理を行なう。まず、前方トレース抽出手段110は、第1の実施の形態における受信先トレース抽出手段80と同様に並列分散システム上で動作する各プロセスから採取したトレースファイル6-1~6-nを全て収集し、収集したトレースファイル6-1~6-nからメッセージ送信処理及びメッセージ送信処理を行なったトレースを抽出してメッセージ送受信リストを作成する（ステップ1001~1003）。

【0061】作成したメッセージ送受信リストを確認し、もし指定されたタスクからメッセージ送信処理が行なわれていた場合（ステップ1004）、そのタスクから送信したメッセージのメッセージ番号を抽出する（ステップ1005）。

【0062】次に、指定したタスクのタスク番号及び抽出した送信メッセージのメッセージ番号と、メッセージ送受信リストの中のメッセージ受信トレース一覧から、送信元のタスク番号及び受信したメッセージのメッセージ番号がそれぞれ一致するタスクを検索する（ステップ1006）。そして、上記指定したタスクと検索されたタスクのトレースをトレースファイル6-1~6-n抽出する（ステップ1007）。

【0063】もし、抽出したタスクにおいてさらにメッセージ送信処理が行なわれていた場合、さらに上述したトレースの抽出処理を再帰的に行ない、メッセージ送信処理が行なわれなくなるまで継続する。これにより、トレース表示手段90は、前方トレース抽出手段110で抽出したタスクのトレースを表示する。

【0064】この結果、並列分散システムで動作するプログラムのデバッグを行なう際、膨大なトレースデータの中から、指定したタスク及びそのタスクからメッセージ送信が行なわれた事が起因となって、連鎖的に実行された全てのタスクのトレース情報のみを表示することができる。

できる。

【0065】次に、本発明における発明の第4の実施の形態に係るデバッグ装置について説明する。図11は、本発明における第4の実施の形態に係るデバッグ装置の構成を示すブロック図である。

【0066】本実施の形態は、第1の実施の形態と同様にタスク実行部1-1~1-n、メッセージ番号生成手段2-1~2-n、通常トレース採取手段3-1~3-n、メッセージ送信手段4-1~4-n、メッセージ受信手段5-1~5-n、トレースファイル6-1~6-n、タスク番号生成手段70、トレース表示手段90を備え、さらに、第1の実施の形態の受信先トレース抽出手段80の代わりに、指定されたタスクを実行させた起因となるタスクのトレースを抽出する後方トレース抽出手段120を備えて構成されている。

【0067】第4の実施の形態による詳細な動作を図に基づいて説明する。図12は、後方トレース抽出手段120の動作を示すフローチャートである。その他の手段の動作については、第1の実施の形態の場合と同様である。

【0068】後方トレース抽出手段120は、デバッグを行なう対象となるタスク番号を指定されて処理を行なう。まず、後方トレース抽出手段12は、受信先トレース抽出手段80と同様に並列分散システム上で動作する各プロセスから採取したトレースファイル6-1~6-nを全て収集し、収集したトレースファイルからメッセージ送信処理トレース及びメッセージ受信処理トレースを抽出してメッセージ送受信リストを作成する(ステップ1201~1203)。

【0069】作成したメッセージ送受信リストを確認し、もし指定されたタスクがメッセージ受信により実行されていた場合(ステップ1204)、そのメッセージを送信した送信元のタスク番号をメッセージ受信トレースから特定する(ステップ1205)。次に、指定されたタスク及び特定したタスク番号に該当するタスクのトレースをトレースファイル6-1~6-nの中から抽出する(ステップ1206)。

【0070】もし、特定したタスクがさらにメッセージ受信によって実行されていた場合、上述したトレース抽出処理を再帰的に行ない、メッセージ受信以外の何らかの原因で実行されたタスクであると判定されるまでタスクのトレースを抽出する。これにより、トレース表示手段90は、後方トレース抽出手段120で抽出したタスクのトレースを表示する。

【0071】この結果、並列分散システムで動作するプログラムのデバッグを行なう際、膨大なトレースデータの中から指定したタスクが実行された起因となったタスクのトレース情報のみを表示することができる。

【0072】上記第1から第4の実施の形態においては、4種類のトレース抽出手段について説明したが、各

々の実施の形態のトレース抽出手段を選択的に組合わせて構成することも可能である。ここで、第5の実施の形態として、第1から第4の実施の形態におけるトレース抽出手段を組合わせたデバッグ装置について説明する。図13は、本発明における第5の実施の形態に係るデバッグ装置の構成を示すブロック図である。

【0073】この実施の形態では、タスク実行部1-1~1-n、メッセージ番号生成手段2-1~2-n、通常トレース採取手段3-1~3-n、メッセージ送信手段4-1~4-n、メッセージ受信手段5-1~5-n、トレースファイル6-1~6-n、タスク番号生成手段70、トレース表示手段90を備え、さらに、第1の実施の形態における受信先トレース抽出手段80と、第2の実施の形態における送信元トレース抽出手段100と、第3の実施の形態における前方トレース抽出手段110と、第4の実施の形態における後方トレース抽出手段120からなるトレース抽出手段130を備え、これらの各抽出手段を選択的に実行できるように構成されている。

【0074】各抽出手段の動作については、図6、図8、図10及び図12に示した内容と同様である。このように構成することで、指定されたタスクからメッセージ送信が行なわれたことにより、それを受けて実行されたタスクのトレース、指定されたタスクをメッセージ送信によって実行させた送信元のタスクのトレース、指定されたタスクからメッセージ送信が行なわれたことにより、そのタスクから連鎖的に実行された全てのタスクのトレース及び指定されたタスクをメッセージ送信によって実行させた送信元のタスクのトレースを遡って、そのタスクを実行させた根源となるタスクのトレースを選択的に抽出することが可能となる。この実施の形態以外に、4種類のトレース抽出手段から2つ以上を任意に選択して組合わせることも可能である。

【0075】なお、本発明は上述した実施の形態に限定されるものではなく、その技術思想の範囲内において様々な変形して実施することができる。

【0076】

【発明の効果】以上説明したように、本発明のデバッグ装置及びデバッグ方法によれば、タスクのタスク番号、メッセージ送信トレースのメッセージ番号、メッセージ受信トレースのメッセージ番号と送信元のタスク番号に基づいて、所定のタスクを特定することにより、タスクのトレースを記録されたトレースの中から抽出するので、多数のプロセスから記録された膨大なトレースデータの中から特定のタスクのトレースのみを抽出して表示することができるので、デバッグ効率が大幅に向上する。

【0077】請求項3及び請求項9の本発明によれば、多数のプロセスから記録された膨大なトレースデータの中から指定されたタスクよりメッセージを受けて実行さ

れたタスクのトレースのみを抽出して表示することができるので、デバッグ効率は大幅に向上する。

【0078】請求項4及び請求項10の本発明によれば、注目したタスクを実行させたメッセージ送信元のタスクのトレースのみを膨大なトレースの中から抽出して表示することができるのでデバッグ効率が大幅に向上する。

【0079】請求項5及び請求項11の本発明によれば、指定されたタスクが実行されたことにより、連鎖的に実行された全てのタスクのトレースのみを膨大なトレースの中から抽出して表示することができるのでデバッグ効率が大幅に向上する。

【0080】請求項6及び請求項12の本発明によれば、指定されたタスクが実行された起因となるタスクのトレースを膨大なトレースの中から抽出して表示することができるので、デバッグ効率が大幅に向上する。

【0081】請求項7及び請求項13の本発明によれば、指定されたタスクからメッセージ送信が行なわれたことにより、それを受けて実行されたタスクのトレース、指定されたタスクをメッセージ送信によって実行させた送信元のタスクのトレース、指定されたタスクからメッセージ送信が行なわれたことにより、そのタスクから連鎖的に実行された全てのタスクのトレース及び指定されたタスクをメッセージ送信によって実行させた送信元のタスクのトレースを遡って、そのタスクを実行させた根源となるタスクのトレースを選択的に抽出することが可能となり、これによりデバッグ効率がより向上する。

【図面の簡単な説明】

【図1】 本発明における第1の実施の形態に係るデバッグ装置の構成を示すブロック図である。

【図2】 トレースのデータ構造を示す図である。

【図3】 第1の実施の形態における通常トレース採取手段の動作を説明するフローチャートである。

【図4】 第1の実施の形態におけるメッセージ送信手段の動作を説明するフローチャートである。

【図5】 第1の実施の形態におけるメッセージ受信手段の動作を説明するフローチャートである。

【図6】 第1の実施の形態における受信先トレース抽出手段の動作を示すフローチャートである。

【図7】 本発明における第2の実施の形態に係るデバッグ装置の構成を示すブロック図である。

【図8】 第2の実施の形態における送信元トレース抽出手段の動作を説明するフローチャートである。

【図9】 本発明における第3の実施の形態に係るデバッグ装置の構成を示すブロック図である。

【図10】 第3の実施の形態における前方トレース抽出手段の動作を説明するフローチャートである。

【図11】 本発明における第4の実施の形態に係るデバッグ装置の構成を示すブロック図である。

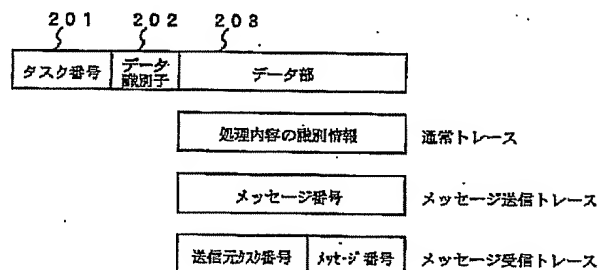
【図12】 第4の実施の形態における後方トレース抽出手段の動作を説明するフローチャートである。

【図13】 本発明における第5の実施の形態に係るデバッグ装置の構成を示すブロック図である。

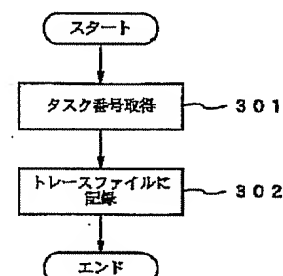
【符号の説明】

- 1-1～1-n タスク実行部
- 2-1～2-n メッセージ番号生成手段
- 3-1～3-n 通常トレース採取手段
- 4-1～4-n メッセージ送信手段
- 5-1～5-n メッセージ受信手段
- 6-1～6-n トレースファイル
- 70 タスク番号生成手段
- 80 受信先トレース抽出手段
- 90 トレース表示手段
- 100 送信元トレース抽出手段
- 110 前方トレース抽出手段
- 120 後方トレース抽出手段
- 130 トレース抽出手段

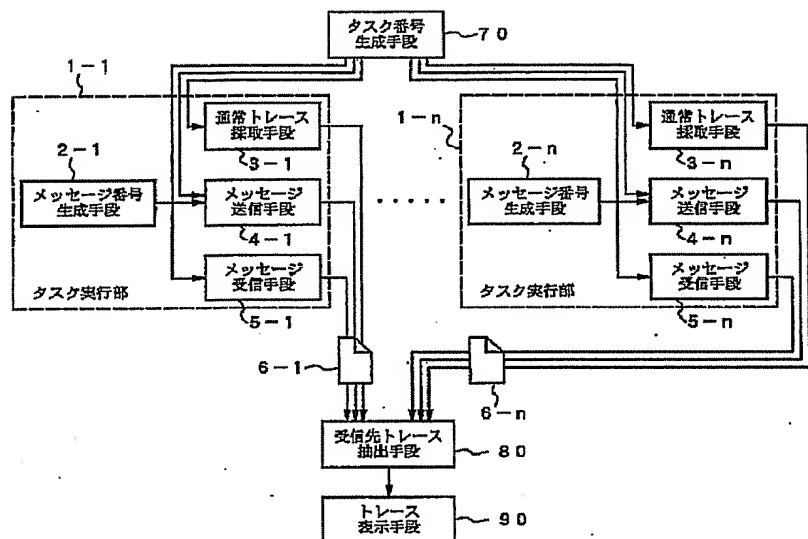
【図2】



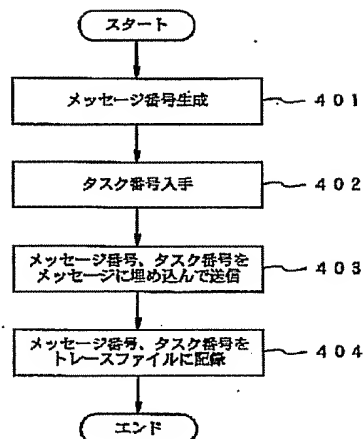
【図3】



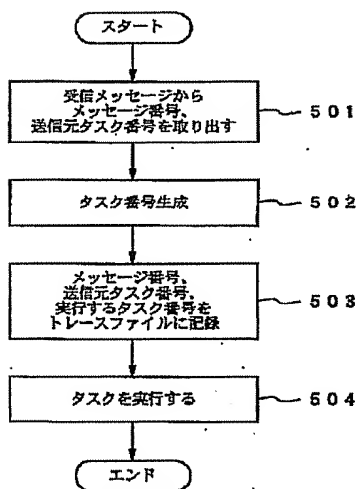
【図1】



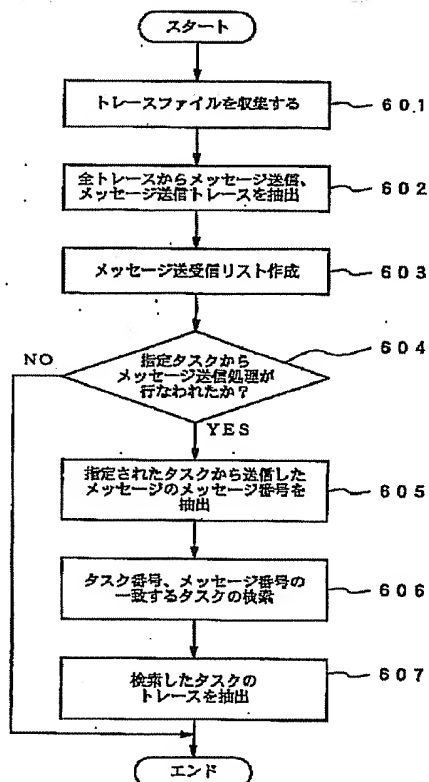
【図4】



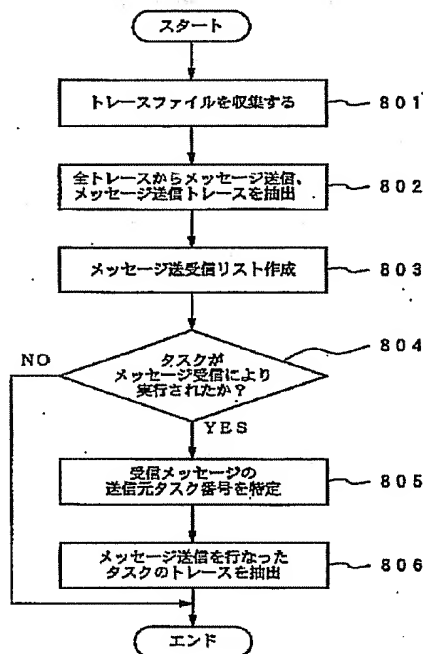
【図5】



【図6】



【図8】



[illegible]

Figure 1 is a block diagram illustrating a task execution system. The system consists of multiple task execution units (1-1 to 1-n) and a central processing unit (70).

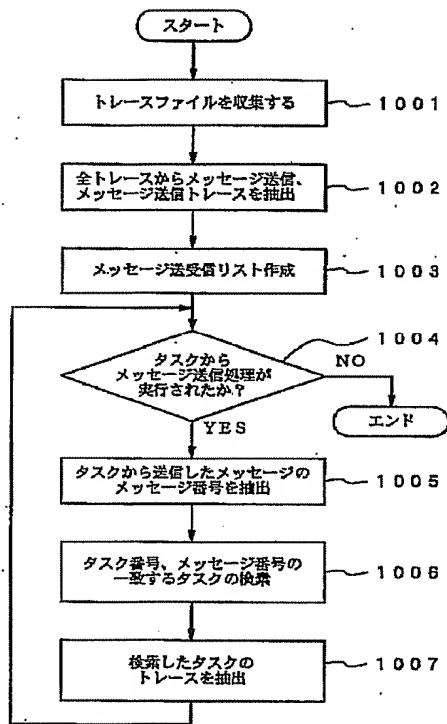
The task execution units (1-1 to 1-n) are organized as follows:

- Task Execution Unit 1-1:**
 - Message Number Generation Step (2-1)
 - Task Execution Section (タスク実行部)
 - Task Trace Extraction Step (3-1)
 - Message Transmission Step (4-1)
 - Message Reception Step (5-1)
- Task Execution Unit 1-n:**
 - Message Number Generation Step (2-n)
 - Task Execution Section (タスク実行部)
 - Task Trace Extraction Step (3-n)
 - Message Transmission Step (4-n)
 - Message Reception Step (5-n)

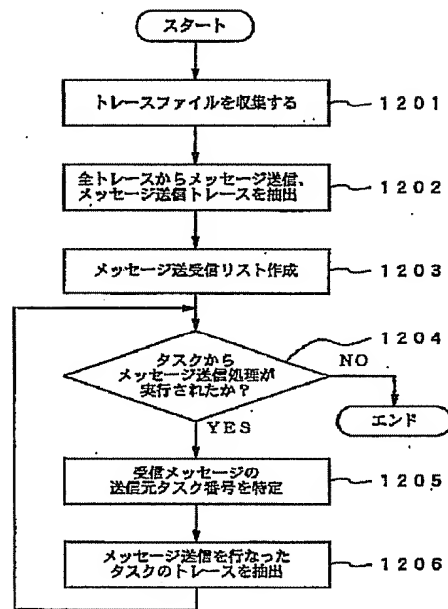
The central processing unit (70) is the Task Number Generation Step (タスク番号生成手段). It provides task numbers to the Message Number Generation Steps (2-1 to 2-n).

The Message Reception Steps (5-1 to 5-n) output data to the Front Trace Extraction Step (6-1 to 6-n). The Front Trace Extraction Steps (6-1 to 6-n) output data to the Front Trace Extraction Step (110). The Front Trace Extraction Step (110) outputs data to the Trace Display Step (90).

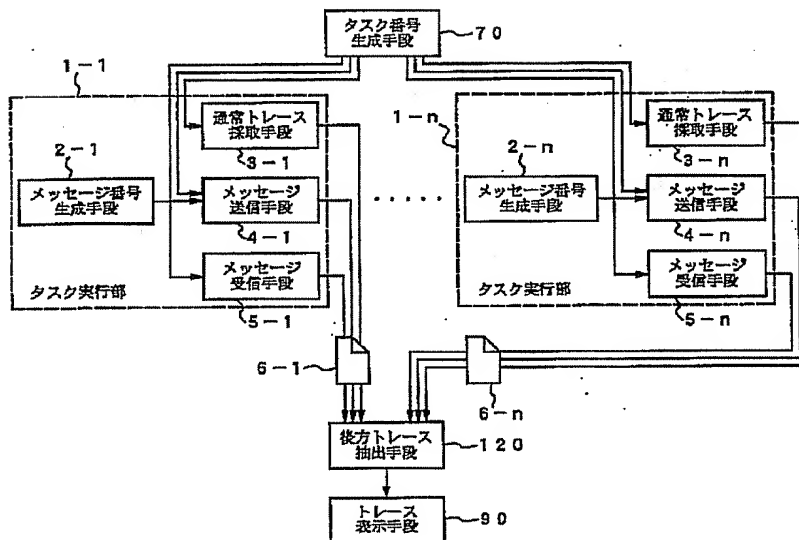
【図10】



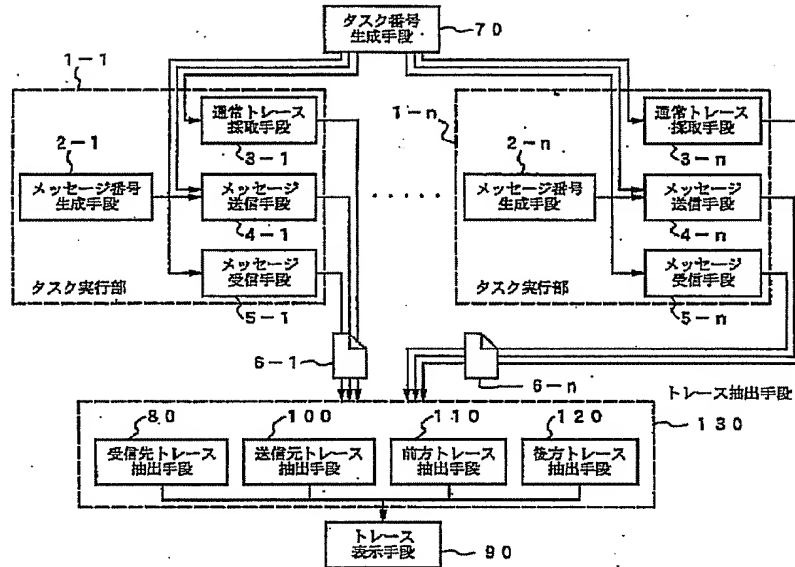
【図12】



【図11】



【図13】



フロントページの続き

(72) 発明者 相場 雄一
東京都港区芝5丁目7番1号 日本電気株
式会社内

Fターム(参考) 5B042 FB08 FB10 FD11 FD36
5B045 BB47 GG01 JJ49
5B089 AA03 AA05 AA07 AA20 AD01
AE08 AF00 CB02 CB03 CC03
EA04
5B098 GA04 JJ07

